

AIF2 Low-level Driver

Release Notes

Applies to Product Release: 01.02.00.02:
Publication Date: July, 2018

Document License

This work is licensed under the Creative Commons Attribution-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Contributors to this document

Copyright (C) 2011 Texas Instruments Incorporated - <http://www.ti.com/>

Texas Instruments, Incorporated
5 Chemin des Presses, 4 allée Technopolis
06800 Cagnes sur Mer,
FRANCE



Contents

Overview	4
AIF2 LLD Dependencies.....	4
New/Updated Features and Quality.....	4
Release 1.2.0.02.....	4
Release 1.2.0.01.....	4
Release 1.2.0.00.....	5
Release 1.1.0.06.....	5
Release 1.1.0.05.....	5
Release 1.1.0.04.....	5
Release 1.1.0.03.....	5
Release 1.1.0.02.....	6
Release 1.1.0.01.....	6
Release 1.1.0.00.....	6
Release 1.0.0.17.....	6
Release 1.0.0.16.....	6
Release 1.0.0.15.....	6
Release 1.0.0.14.....	6
Release 1.0.0.13.....	6
Release 1.0.0.12.....	6
Release 1.0.0.11.....	7
Release 1.0.0.10.....	7
Release 1.0.0.09.....	7
Release 1.0.0.08.....	7
Release 1.0.0.07.....	7
Release 1.0.0.06.....	7
Release 1.0.0.05.....	7
Release 1.0.0.04.....	7
Release 1.0.0.03.....	7
Release 1.0.0.02.....	7
Release 1.0.0.00.....	7
Release 0.0.0.00.....	8

Resolved Incident Reports (IR).....	8
Known Issues/Limitations	8
Licensing.....	8
Delivery Package	8
Instructions to upgrade the Ild	9
Instructions to rebuild the Ild	9
Instructions to create and build the Ild test examples	9
Directory structure.....	9
AIF2 LLD build, EVM setup and debug setup	10
Building the AIF2 LLD and test projects	10
TMDXEVM6670L/LE hardware configuration for AIF2 testing	10
Running AIF2 test projects from the debugger.....	10
AIF2 LLD usage	11
AIF2 LLD test projects	11

Release Notes

AIF2 LLD version 01.02.00.02

Overview

This document provides the release information for the latest AIF2 low-level driver which should be used by drivers and application that interface with AIF2, QMSS and CPPI IPs.

This AIF2 low-level driver aims at generalizing the configuration of AIF2 for different modes (CPRI/OBSAI/ Generic packet, WCDMA/LTE/Dual mode).

It should be noted that AIF2 LLD support a single instance of the driver run from only one of the KeyStone SOC cores. An AIF2 LLD test example showing how to use a single LLD instance across multiple cores is provided in the AIF2 LLD package.

AIF2 LLD includes:

- Compiled library (Big and Little) Endian of AIF2 low-level driver.
- Source code.
- A set of AIF2 LLD tests
- API reference guide
- User documentation details

AIF2 LLD Dependencies

AIF2 LLD is dependent on following external components delivered in PDK package:

- CSL (from PDK)
- QMSS LLD (from PDK)
- CPPI LLD (from PDK)
- EDMA3 LLD (for the dual mode test framework)

New/Updated Features and Quality

This is an official release, tested by the system and applications team. Processor SDK RTOS 5.00.00 is used during the validation phase.

Release 1.2.0.02

- Updated buildlib.xs to add RULES_MAKE macro to support build based on custom Rules.make

Release 1.2.0.01

- RTSC scripts: add compile time define to include correct keystone I/II device CSL headers.
- Test projects

- rename test project name to align with new pdkProjectCreate rule
- update test utility files to align with Proc-SDK CSL changes

Release 1.2.0.00

- LLD user mode driver on ARM Linux: this required importing the AIF2 CSL functional layer into the driver. Therefore the LLD has now a new layer, called AIF2 FL, allowing access to the AIF2 CSL register layer (check for details in the Architecture chapter of the LLD user's guide).

The application/customer code using LLD needs to be migrated using the following rules:

- Functional layer header files need to be migrated to:
 - `#include <ti/csl/csl_aif2.h>` to `#include <ti/drv/aif2/aif2fl.h>`
 - `#include <ti/csl/csl_aif2GetHwStatusAux.h>` to `#include <ti/drv/aif2/aif2fl_getHwStatusAux.h>`
 - `#include <ti/csl/csl_aif2HwControlAux.h>` to `#include <ti/drv/aif2/aif2fl_hwControlAux.h>`
 - `#include <ti/csl/csl_aif2HwSetupAux.h>` to `#include <ti/drv/aif2/aif2fl_hwSetupAux.h>`
- Functional layer functions: all functions have an equivalent renamed one such as `CSL_aif2GetVersion()` renamed to `Aif2Fl_getVersion()`
- Functional layer data types: all data types have an equivalent renamed one such as `CSL_Aif2PidStatus` to `Aif2Fl_PidStatus`
- Functional layer definitions: all definitions have an equivalent renamed one such as `CSL_AIF2_LINK_PROTOCOL_CPRI` to `AIF2FL_LINK_PROTOCOL_CPRI`
- Functional layer handle in the AIF2 driver instance was renamed:


```
AIF_ConfigObj aifObj; // Main AIF2 LLD object for AIF2 configuration
aifObj.hCsl to aifObj.hFl
```
- Support for AxC using different sampling rates on the same CPRI link. The scenarios verified for that new "multiRate" link property are: LTE20+LTE10, LTE10+LTE5, LTE20+LTE5, LTE+WCDMA.

Release 1.1.0.06

- Add initial support for LLD user mode driver on ARM Linux
- Updated RTWP header + code in the LLD test utilities

Release 1.1.0.05

- Addition OSAL functions for RM module
- Adding missing pragmas for code and data to help memory placement of the Aif2 driver
- Fix on physyncsel, radsyncsel, and phytcompvalue combinations

Release 1.1.0.04

- API enhancement to support LLD integration with AZ WCDMA PHY
- Changed the procedure to enable AIF2 data trace to match the factory apps recommended one
- Fix for SDOCM00111714: multicore example build issue for Linux
- Added support for CPRI LTE40 in ltecheckrf test and OBSAI in wcdmacheckrf test

Release 1.1.0.03

- Further enhancements of ltecheckrf.c test to support new K-II+Drboc+Marconi radio kits
- Fix for SDOCM00088973: `CSL_BootCfgLockKicker()` should be removed from all LLD examples
- Fix for Generic packet mode when using other links than link 0
- Fix for Cpri relay with TCI6614 at 1198.08 MHz setting to comply with 1.2 GHz set by u-boot

Release 1.1.0.02

- Further enhancements of ltecheckrf.c test to support new K-II+Drboc+Marconi radio kits
- Bug fixes (Coverity issues)

Release 1.1.0.01

- Further enhancements of ltecheckrf.c test to support new K-II+Drboc+Marconi radio kits
- API enhancement to support LLD integration with TI LTE PHY
- Fix for trace data capture for Lte Obsai testcases
- Support for Cpri/Obsai Lte 15MHz
- LTE multicore test example enhanced from 4 to 8 cores for KeyStone-II devices
- Modified AIF_init.c to support KeyStone-II AIF2 SerDes CSL fix for B4 macro Shutdown/Restore

Release 1.1.0.00

- Added a LTE multicore test example
- Added support for RM (K-II) in test utilities when setting up Qmss and Cppi (mnavUtils.c)
- Enhanced ltecheckrf.c test to support new K-II+Drboc+Marconi radio kits
- Modified AIF_init.c for K-II devices to support AIF2 SerDes CSL API changes

Release 1.0.0.17

- Bug fixes

Release 1.0.0.16

- Fix to support AIF2 and SerDes SW reset on KeyStone-II devices
- Sw workaround implementation for KeyStone-I and KeyStone-II “AIF2 CPRI Fast C&M Restrictions” Usage Note. This is done in cprifastcm test folder.
- Added support for AIF2 data trace (used in Obsai mode)
- Wcdma Dual Carrier Check RF tests working with EVM + SCBP (CPRI RELAY)
- Support for disabling each individual PE or PD channels (new APIs)
- Modified RTWP_evm.c to make it easy to disable sanity checks and debug info collection.

Release 1.0.0.15

- Fix to support all test projects in big endian mode
- KeyStone-II **migration**: dual mode test

Release 1.0.0.14

- Fix for OBSAI 8x rate
- Fix for Linux conflict when running AIF2LLD test examples
- Cpri relay code (EVMTCI6614/SCBP): added TCI6614 sync auto-detection of CPU clock (1.228G or 983M)
- KeyStone-II **migration**: fast c&m, Lte/Wcdma single-tone and Generic packet tests (dual mode test under debug)

Release 1.0.0.13

- Added support for KeyStone-II xdc platform: ti.platforms.evmTCI6638K2K

Release 1.0.0.12

- Added support for KeyStone-II SerDes CSL

- Added support for CPRI LTE TDD
- Added support for OBSAI LTE FDD
- Lte Tdd Check RF tests working with EVM + SCBP (CPRI RELAY)

Release 1.0.0.11

- KeyStone-II **migration**: Lte and Wcdma Check RF tests working with EVM + SCBP (CPRI RELAY)

Release 1.0.0.10

- KeyStone-II **migration**: initial Lte and Wcdma tests working with EVM

Release 1.0.0.09

- Added Cpri RF check test projects for both Lte and Wcdma. These tests were primarily designed to work with Cpri relay setups or Scbp+Tsw3736 standalone setups
- LLD migrated to KeyStone-II devices (TCI6634/8K2K, TCI6636K2H), initial Lte and Wcdma tests for associated simulators
- Superpacket workaround optional for Cpri Lte Ingress antenna carriers (C6670 advisory 12)

Release 1.0.0.08

- Adding support Vendor Specific and Fast C&M independently for both WCDMA and LTE.
- Modified cprifastcm test case to combine Lte and Fast C&M and Vendor specific traffic.
- Adding support for Lte 5MHz in cpri relay and single tone test projects

Release 1.0.0.07

- Adding support for Lte 10MHz in cpri relay and single tone test projects

Release 1.0.0.06

- Support for Dual mode using the DIO engine approach
- Cpri relay Lte test sends a single tone signal on the first antenna and a dual tone signal on the second antenna

Release 1.0.0.05

- Support for Azcom SCBP single tone tests based on SYS/BIOS (both Wcdma and Lte)

Release 1.0.0.04

- Bug fix for Cpri fast C&M

Release 1.0.0.03

- Bug fixes and enhancements for Small Cell Cpri relay solution.
- Support for Fdd Lte 5 and 10 MHz configurations

Release 1.0.0.02

- Port to TCI6614. C6670 and TCI6614 use the same AIF2 LLD codebase.
- Support for LTE flexible antenna carrier sample packing for the multiple antenna carrier use-cases. With this, addition of super packet workaround for C6670 Advisory 12.

Release 1.0.0.00

- Support for the following AIF2 hardware configuration modes:

- WCDMA DL/UL, CPRI 4x and 8x modes
- WCDMA DL/UL, OBSAI 4x mode
- LTE DL/UL, CPRI 4x mode
- Generic packet, CPRI/OBSAI 4x mode
- Link retransmissions and daisy chaining
- Example tests on C6670/TCI6616 EVMs from both Advantech (default) and Lyrtech.
- Main tested features
 - Wcdma Downlink/Uplink for up to 4 links
 - Wcdma Downlink/Uplink with link Retransmitter
 - LTE Downlink/Uplink on multiple links with multiple carriers
 - Generic packet on multiple links.

Release 0.0.0.00

- Support for the following AIF2 hardware configuration modes:
 - WCDMA, CPRI 4x mode, both little and big endian
 - WCDMA, OBSAI 4x mode, both little and big endian
 - Legacy ABT (Antenna bursty traffic) mode, AIF2 to AIF2, little endian
- Example tests on C6670/TCI6616 EVMs from both Advantech (default) and Lyrtech.
- Main tested features
 - Downlink/Uplink for up to 4 links using same DIO engine
 - Downlink/Uplink for several links on different DIO engines
 - CPRI control words using the multicore navigator
 - ABT mechanism up to 4 independent links.
 - ABT Performance (throughput and cpu load)

Resolved Incident Reports (IR)

IR Parent/ Child Number	Severity Level	IR Description
PRSDK-2194	NA	Add RTOS Installer script to autosest SDK_INSTALL_PATH

Known Issues/Limitations

Licensing

Please refer to the software Manifest document for the details.

Delivery Package

AIF2 LLD is delivered as part of PDK packages for C6670, TCI6614, and KeyStone-II.

Instructions to upgrade the lld

The AIF2 LLD can be unzipped anywhere. It follows the PDK directory structure though and could be unzipped on top of the PDK, so that the LLD can be found under [your pdk package path]/ti/drv/aif2

Now, if a new version of the lld (tar file) is available aside from a complete MCSDK/PDK package, follow these steps to upgrade the lld in your pdk environment:

- Copy the tar file to `$your_pdk_path\packages`
- Rename `$your_pdk_path\packages\ti\drv\aif2` folder
- Untar `aif2_lld_new.tar` and it should recreate `$your_pdk_path\packages\ti\drv\aif2` folder

Instructions to rebuild the lld

The PDK package comes with a pre-built version of the LLD. To rebuild all PDK components, please refer to the following instructions:

http://processors.wiki.ti.com/index.php/BIOS_MCSDK_2.0_User_Guide#Building_CSL_and_the_Low_Level_Device_Drivers.

Still, it is possible to only rebuild the LLD. For this, open a command prompt and go to

`$your_pdk_path\ti\drv`. Then call `pdksetupenv.bat`, and rebuild the AIF2 LLD:

```
gmake -C aif2 LIBDIR=./lib clean
gmake -C aif2 LIBDIR=./lib all
```

Instructions to create and build the lld test examples

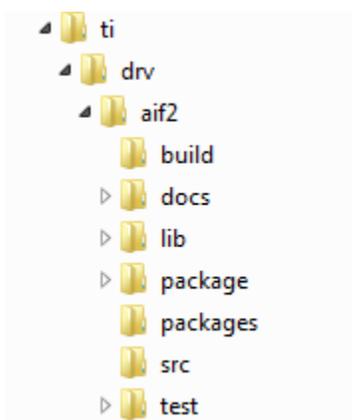
The PDK example projects are located under “\packages\ti\drv\exampleProjects” directory. This is actually a CCS Eclipse workspace with all driver examples and tests. Open this workspace in CCS and import the macros.ini file provided in the driver package (CCS File menu option->Import->CCS->Managed Build Macros). The PDK examples can now be built (check the project configurations of your interest first).

If the LLD was upgraded, you may need to recreate a workspace with driver examples, this is possible by:

- Edit `pdkProjectCreate.bat` file from the pdk installation and set the `CCS_INSTALL_PATH` properly (CCS version needs to be compatible from PDK package)
- Open a dos command and call `pdksetupenv.bat`
- Call `pdkProjectCreate.bat`. That will create a CCS Eclipse workspace called `MyExampleProjects`

Directory structure

The following is the directory structure after the AIF2 LLD package has been installed:



Directory name	Description
ti/drv/aif2	The top level directory contains the following: <ul style="list-style-type: none"> - Readme file for required environment variables - Eclipse standard makefile project that allows re-building the LLD - Header files which are provided by the AIF2 LLD and should be used by the application developers for driver customization and usage
ti/drv/aif2/docs	The directory contains the AIF2 LLD documentation <ul style="list-style-type: none"> - This includes this release notes + user documentation details - The generated API reference guide
ti/drv/aif2/lib	The "lib" folder has pre-built Big and Little Endian libraries <ul style="list-style-type: none"> - ti.drv.aif2.ae66(e) for the release versions
ti/drv/aif2/src	Source code for the AIF2 LLD
ti/drv/aif2/test	The "test" directory in the AIF2 LLD has unit test cases which are used by the system and applications team to test the AIF2 LLD for the different supported modes. Please note that these tests are only a subset of the test suite used by the AIF2 LLD team

AIF2 LLD build, EVM setup and debug setup

The LLD uses a CCSv5 based environment for building the LLD itself and its test executables. The LLD is built using the XDC methodology, which can either be invoked from an Eclipse standard make project or from the command line with the xdc command launched from the root of the LLD package.

Building the AIF2 LLD and test projects

Please refer to the ReleaseNotes_PDK.pdf in PDK/docs folder. It contains information on build environment and execution of PDK test examples.

TMDXEVM6670L/LE hardware configuration for AIF2 testing

AIF2 loopback tests: only single EVM setup is required with default EVM settings.

AIF2 DSP-to-DSP tests: 2 modified-rev2 or rev3 EVMs connected to a CI break-out connector (BOC) are required. The BOC configuration is as such: J1 open, J2 open, J3: ON-ON-ON, J4: OFF-ON-ON, J5: ON-ON-ON-ON-ON-OFF, J6: 2-3 connected (board B powers the BOC), shunts on JP1/JP2/JP3, JP4/JP5/JP6/JP7/JP8 open, JP9 (timer output) connected to board A 80-pin connector pin 57 with the grey wire, JP10 open. The modified-rev2 EVM configuration is as such: SW3: 1-off, 2-off, 3-on, 4-off, SW4, SW5, SW6, SW9 : all on COM_SEL1: 1-3, 2-4, UART over USB (if needed to verify common REFCLK locked from BOC via UART terminal)

Running AIF2 test projects from the debugger

To debug with Code Composer Studio v5, use a ccxml configuration for the TMS320C6670 EVM and attach "%CCSV5_ROOT%\ccs_base\emulation\boards\evmc6670l\gel\evmc6670l.gel" from your CCS installation folder to each of the C6670 CPU cores. When testing in loopback mode, only single DSP ccxml configuration is required. When testing AIF2-AIF2 communication between 2 DSPs, either 2-DSP ccxml configuration is required if both DSP are in the same JTAG scan chain, or 2 1-DSP configurations is required if each of the DSP has its separate JTAG scan chain. When using 2 TMS320C6670 EVMs with a CI break-out connector, both options are available.

To run tests, you'll need to load the appropriate test executable depending on your setup configuration. For each of the test projects, you find 3 project configurations: Advantech_EVM_DSP1, Advantech_EVM_DSP2, and Lyrtech_EVM. The Advantech executables are the ones to use together with the TMDXEVM6670L/LE board. DSP1 executable can be used for single DSP tests or dual DSP tests. In case of dual DSP tests, DSP_1 corresponds to the EVM (usually board A) that connects DSP timer0 output signal to AIF2 radsync/physync signals of the 2

EVMs. DSP_2 needs to be loaded and run first, and waits for DSP_1 to trigger an external frame synchronization for the AIF2 radio timers on both DSPs.

AIF2 LLD usage

Please go to the AIF2 user's guide for usage details.

AIF2 LLD test projects

The directory [your path]/aif2/test contains the AIF2 LLD test suite and some CSL utilities. Those can be used as examples for the LLD user. The test suite consists of the following tests:

aif2WcdmaTestProject : this a CPRI/OBSAI DL/UL WCDMA 4x test.

aif2LteTestProject : this a CPRI LTE test.

aif2GenericTestProject : this a CPRI Generic packet test.

aif2LteCheckRfTestProject : This CPRI LTE Check RF test runs on a single link of AIF2 and configures 2 AxCs. Both AxCs IQ data are pure single or dual tone sinewaves at given frequencies, such that upon reception of LTE symbols for both antennas, IQ data can be checked with fft software. If the signal doesn't go from digital to analog and back, it can also be checked bitwise. This CPRI LTE Check RF test is designed to work either in internal loopback mode, or with a dual EVM setup connected with a break-out card, or with a CPRI Relay setup (tci6614 evm + scbp + tsw3726), or with a Scbp/Tsw3726 standalone setup.

aif2LteTddCheckRfTestProject : This CPRI LTE TDD Check RF test runs on a single link of AIF2 and configures 2 AxCs. Both AxCs IQ data are pure single or dual tone sinewaves at given frequencies, such that upon reception of LTE symbols for both antennas, IQ data can be checked with fft software. This CPRI LTE TDD Check RF test is designed to work either in internal loopback mode, or with a dual EVM setup connected with a break-out card, or with a CPRI Relay setup (tci6614 evm + scbp + tsw3726), or with a Scbp/Tsw3726 standalone setup.

aif2WcdmaCheckRfTestProject : This CPRI WCDMA Check RF test runs on a single link of AIF2 and configures 2 AxCs. Both AxCs IQ data are pure single or dual tone sinewaves at given frequencies, such that upon reception of Wcdma frames for both antennas, IQ data can be checked with fft software. If the signal doesn't go from digital to analog and back, it can also be checked bitwise. This CPRI WCDMA Check RF test is designed to work either in internal loopback mode, or with a dual EVM setup connected with a break-out card, or with a CPRI Relay setup (tci6614 evm + scbp + tsw3726), or with a Scbp/Tsw3726 standalone setup.

aif2CpriFastCMTestProject : this a CPRI/LTE 4x test that includes transmission and reception of Cpri fast C&M packets and vendor specific information.

aif2LteSingleToneTestProject : this is a LTE tests that uses the Aif2 CPRI configuration from TI Small Cell SW product and exercises the RF chain (Azcom SCBP + TI TSW3725) using a single tone signal so that it can be clearly observed on a spectrum analyzer. This test demonstrates how to use SYS/BIOS with AIF2 LLD.

aif2WcdmaSingleToneTestProject : this is a WCDMA tests that uses the Aif2 CPRI configuration from TI Small Cell SW product and exercises the RF chain (Azcom SCBP + TI TSW3725) using a single tone signal so that it can be clearly observed on a spectrum analyzer. This test demonstrates how to use SYS/BIOS with AIF2 LLD.

aif2DualModeTestProject : this is a dual mode (simultaneous Lte and Wcdma antenna traffic) test that uses the DIO approach for both traffic types and reconstructs Lte symbol packets on the ingress side using EDMA3 LLD. This test demonstrates how to use SYS/BIOS with AIF2 LLD.